



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/286,133	04/01/1999	STEWART SABADELL	49658-024	4582

29989 7590 03/11/2004

HICKMAN PALERMO TRUONG & BECKER, LLP
1600 WILLOW STREET
SAN JOSE, CA 95125

EXAMINER

SHARON, AYAL I

ART UNIT	PAPER NUMBER
----------	--------------

2123

29

DATE MAILED: 03/11/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application N

09/286,133

Applicant(s)

SABADELL ET AL.

Examiner

Ayal I Sharon

Art Unit

2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 22 January 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-13 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 14-17 is/are allowed.
- 6) ☒ Claim(s) 1-13 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.

DETAILED ACTION

Introduction

1. In view of the Appeal Brief filed on 01/22/04, PROSECUTION IS HEREBY REOPENED. The rejections of Claims 1-13 are set forth below.

To avoid abandonment of the application, Applicants must exercise one of the following two options:

- (1) file a reply under 37 CFR 1.111 (to this non-final Office action); or,
- (2) request reinstatement of the appeal.

If reinstatement of the appeal is requested, such request must be accompanied by a supplemental appeal brief, but no new amendments, affidavits (37 CFR 1.130, 1.131 or 1.132) or other evidence are permitted. See 37 CFR 1.193(b)(2).

2. Prosecution has been reopened because the claim rejections have been revised in order to provide more detail in the mapping of Applicants' claims onto the Lowry et al. reference.
3. Moreover, two new references have been added to the record to support the Examiner's assertions that the CAD and CAM systems described by Lowry inherently have different functionalities, where the design functionalities of CAD are not supported in CAM, and vice versa. These references are:

- a. Bak, David. "CAD/CAM comes to Windows". Design News. March 23, 1998. p.55.
 - b. Beard, Tom. "Tying It All Together". Modern Machine Shop. Feb 1998. p.90.
4. Claims 14-17 were allowed in paper #27.

Claim Interpretations

5. Examiner interprets "keys" as, for example, being "certain property values in the object entry ... for determining where in the hierarchy tree structure to store the geometry of the object." (specification, p.19, lines 22-26)
6. Examiner interprets "filter objects" as, for example, being used "to determine where the geometry of an object is to be inserted into the tree structure." (specification, p.16, line 25 to p.17, line 3)
7. Examiner interprets "modified stack" as, for example, being "... used to record and maintain some of the modifications that are made to objects through the use of a visual rendering application." (specification, p.23, lines 14-15)
8. Examiner interprets an "object" as, for example, being a CAD or visual rendering software representation of a real-world physical object. "For example, the object properties associated with a visual rendering application typically allow objects to be viewed in a target scene as having a particular texture or material, or to visualize how certain lighting would look when applied to a particular object." (specification, p.4, lines 10-18).

Summary of the Lowry Reference

9. The objective of Lowry et al. ("Lowry") is to simplify "integrating" multiple applications, where "integration" is using applications' output as input for other applications (See col.1, lines 15-43). For example, a CAD and a CAM program receive data from another CAD program (See col.1, lines 20-56). Lowry teaches that time effort and expense increase dramatically with the number of applications that are integrated (col.2, lines 4-9).
10. Lowry uses a common data structure (See Fig.1, Item 140; col.2, lines 9-37; col.27, lines 63-67) that represents the sum total of all of the information accessed by the various application programs (See col.5, lines 32-46). This common data structure is stored in both the master and slave copies of the common database (Fig.5A, Items 510 and 520; and col.9, lines 23-27).
11. In addition to the common database, the various application programs (See Fig.1, Items 130, 132, and 134; Fig. 5A, Items 530, and 560; col.9, lines 29-37) have their own internal databases, which store information either in the common data structure format, or in their own proprietary format (see col.5, lines 32-46).
12. Application programs with proprietary formats (e.g. Fig.5A, Items 530 and 560) input data into the common database via converter programs (Fig.5A, Items 540 and 570) that translate data from the proprietary format to the common format, and then input the data into the shared databases (See col.10, lines 35 to col.11, line 2). There are also retrieval programs (Fig.5A, Items 580 and 590) that retrieve data from the common database, translate them into the proprietary

format, and input the data into the application programs (See col.9, lines 29-52 and col.10, lines 16-26). In addition, there is a Update Synchronizer (Fig.5A, Item 550; col.10, lines 49-60) that manages the updating of the data stored in the common database.

13. The common database uses the ALTER and DBC locks (see col.11, lines 57-64 and col.35, lines 34-41) to ensure that two applications are not simultaneously modifying the same portion of the database, called a "macropage". The locks also ensure that only one user can make modifications to a version of a "macropage" (See col.31, lines 29-51). Each "macropage" is a 64k byte portion of a file, where a file consists of 128 512-byte pages (See col.12, lines 60-66). The DBC keeps track of versions (See col.31, lines 42-51; Figs.30-31).
14. The invention is not limited to a "Source" and a "Target" application, because the objective of the invention is to facilitate the integration of many applications. (See col.2, lines 4-22). In addition, these applications are a combination of CAD and CAM applications (See col.1, line 49 to col.2, lines 4-22). Therefore, the Examiner is mapping the terms in the claims onto Lowry as follows:

- 15. Source and Target Applications:** Application programs 530 and 560 in (Fig.5A; col.9, lines 29-37) have their own internal databases (see col.5, lines 32-46). They both can read from and write to the common database via dedicated retrieval and conversion programs (col.9, lines 29-52; col.10, line 16 to col.11, line 2). The application programs can be either CAD or CAM programs (See

col.1, lines 20-56). **Examiner is interpreting 530 as a source CAD application.**

Examiner is defining 560 as a target CAM application.

16. Source and Target Object: The application programs (See Fig. 5A, Items 530, and 560; col.9, lines 29-37) have their own internal databases, which store information either in the common data structure format, or in their own proprietary format (see col.5, lines 32-46). **Examiner is interpreting the data in source application 530's internal database as being the source object, and the data in the target application 560's internal database as being the target object.**

Claim Rejections - 35 USC § 102

17. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

18. The prior art used for these rejections is as follows:

19. U.S. Patent 4,864,497. (Henceforth referred to as "**Lowry**").

20. The claim rejections are hereby summarized for Applicant's convenience. The detailed rejections follow.

21. **Claims 1-2, 4-10 and 12-13 are rejected under 35 U.S.C. §102(b) as being anticipated by Lowry.**

22. In regards to Claim 1, Lowry expressly teaches the claimed limitations.

- a. The first limitation of Claim 1 states: "generating a source object in a source application"; in comparison, Lowry teaches that "information created in application program 530's database can be sent to master database 510." (See col.10, lines 48-60; col.5, lines 32-46).
- b. The second limitation of Claim 1 states: "translating the source object to a target object in a target application, wherein the target application has a format that is not supported by the source application"; in comparison, Lowry teaches the following:
 - Source application 530 sends the source object to converter program 540. (See col.10, lines 35 to col.11, line 2).
 - Converter program 540 translates the source object to the common format, and then inputs the object into the shared master database 510. (See col.5, lines 32-46; col.10, lines 35 to col.11, line 2).
 - Retrieval program 590 retrieves the object from the shared slave database 520, translates them into the target application 560 format, and inputs the object into the target application 560. (See col.9, lines 29-52 and col.10, lines 16-26).
- c. The third limitation of Claim 1 states: "performing a first modification to the target object, wherein said first modification is not supported by said source application"; in comparison, Lowry teaches the following:

- Lowry teaches integration of CAD and CAM applications (col.1, lines 49-56). Examiner has interpreted target application 560 as being a CAM program.
 - Information created in an application program database can be sent to master database. (See col.10, lines 48-60; col.5, lines 32-46).
 - It is inherent that functions performed in CAM programs are not supported by CAD programs. (See Lowry, col.1, lines 20-56). CAD programs are for component designs, while CAM programs are for automating the machining functions needed to produce the CAD designs. For further evidence of this, see the figure in the reference by David Bak ("CAD/CAM comes to Windows"), as well as the first two pages of the reference by Tom Beard ("Tying It All Together").
- d. The fourth limitation of Claim 1 states: "performing a second modification to said source object in said source application"; in comparison, Lowry teaches the following:
- Lowry teaches integration of CAD and CAM applications (col.1, lines 49-56). Examiner has interpreted source application 530 as being a CAD program.

- Information created in an application program database can be sent to master database. (See col.10, lines 48-60; col.5, lines 32-46).
- e. The fifth limitation of Claim 1 states: “revising said target object in said target application to reflect said second modification to said source object without removing said first modification to said target object”; in comparison, Lowry teaches the following:
- Update synchronizer 550 manages the updating of common database 510 with updated information from application programs 530 and 560. (col.10, lines 48-60).
 - Shared databases 510 and 520 contain all of the information accessed by all the application programs. (See col.5, lines 32-46 and col.10, lines 35-47)
 - ALTER and DBC locks ensure that 530 and 560 are not simultaneously modifying the same portion - “macropage” - of the database (see Fig. 5B; col.11, lines 57-64 and col.35, lines 34-41), and that only one user can make modifications to a version of a “macropage” (See col.31, lines 29-51). Therefore the revisions do not conflict or overwrite one another.
 - Retrieval program 590 retrieves data from the common databases 510 and 520, translates them into the target application 560 format, and inputs the revised target object data

into the target application 560. (See col.9, lines 29-52 and col.10, lines 16-26).

Therefore, Lowry anticipates claim 1.

23. In regards to Claim 2, Lowry expressly teaches the following limitations:

2. The method of Claim 1, wherein the step of performing the first modification to the target object includes the step of performing a type of modification that cannot be performed using said source application.

It is inherent that functions performed in CAM programs are not supported by CAD programs. (See Lowry, col.1, lines 20-56). CAD programs are for component designs, while CAM programs are for automating the machining functions needed to produce the CAD designs. For further evidence of this, see the figure in the reference by David Bak ("CAD/CAM comes to Windows"), as well as the first two pages of the reference by Tom Beard ("Tying It All Together"). Therefore a modification performed in a CAM application cannot be performed in a CAD application, so modifications in target application 560 cannot be performed in source application 530.

24. In regards to Claim 4, Lowry expressly teaches the following limitations:

4. The method of Claim 1, wherein:
the source object is associated with a source geometry and one or more source properties; and
the step of translating the source object to the target object includes the steps of:
translating the source geometry to a target geometry; and
translating the one or more source properties to one or more target properties.

Lowry teaches the following (Lowry: col.5, lines 32-45):

In accordance with the present invention, application programs 130, 132, and 134 share a common data structure 140 which is

based upon an Attributive data model. The Attributive data model represents all of the information accessed by application programs 130, 132 and 134 as various combinations of a single primitive. In general, application programs 130, 132 and 134 have their own databases which contain data either in the Attributive data model's format or in some other format. If the application program information is not in the Attributive data model format, it is converted into that format by means described below. In this manner, data structure 140 can be accessed by each of the application programs 130, 132, and 134.

Lowry also teaches (See col.9, lines 29-52 and col.10, lines 16-26) that Retrieval program 590 retrieves data from the common databases 510 and 520, translates them into the target application 560 format, and inputs the revised target object data into the target application 560. Lowry does not expressly use the terms "target geometry" and "source geometry", however, it is inherent that both CAD and CAM files are comprised of geometrical information.

25. In regards to Claim 5, Lowry expressly teaches the following limitations:

5. The method of Claim 1, wherein the step of translating the source object to the target object includes the step of:

building a mapping based on a translation between the source object and the target object.

Lowry teaches the following (Lowry: col.5, lines 32-45):

In accordance with the present invention, application programs 130, 132, and 134 share a common data structure 140 which is based upon an Attributive data model. The Attributive data model represents all of the information accessed by application programs 130, 132 and 134 as various combinations of a single primitive. In general, application programs 130, 132 and 134 have their own databases which contain data either in the Attributive data model's format or in some other format. If the application program information is not in the Attributive data model format, it is converted into that format by means described below. In this manner, data structure 140 can be accessed by each of the application programs 130, 132, and 134.

Lowry also teaches (See col.9, lines 29-52 and col.10, lines 16-26) that Retrieval program 590 retrieves data from the common databases 510 and 520, translates them into the target application 560 format, and inputs the revised target object data into the target application 560.

26. In regards to Claim 6, Lowry expressly teaches the following limitations:

6. The method of Claim 5, wherein the step of building the mapping includes the step of:

constructing a hierarchical tree structure, wherein the hierarchical tree structure is based on one or more properties associated with the source object.

Lowry teaches that the attribute data objects are mapped hierarchically (Lowry: col.2, line 60 to col.3, line 10. Emphasis Added):

Specifically, the method of creating the data structure comprises the steps, executed by said data processor, of creating a different one of the attribute data objects for each of the node data and organizing the attribute data objects hierarchically according to a being-held relationship by choosing from the attribute data objects for each of the attribute data objects, a holder data object such that a hierarchical being-held relationship exists between each attribute data object and a single holder data object.

27. In regards to Claim 7, Lowry expressly teaches the following

limitations:

7. The method of Claim 6, wherein the source object is associated with a source geometry and one or more source properties; and

the step of constructing the hierarchical tree structure includes the steps of:

generating a set of tree objects, wherein the set of tree objects include one or more filter objects that are based on said source properties;

translating the source geometry to a target geometry; and

inserting said target geometry into said hierarchical tree structure based on said one or more filter objects.

Lowry teaches the following (Lowry: col.5, lines 32-45):

In accordance with the present invention, application programs 130, 132, and 134 share a common data structure 140 which is based upon an Attributive data model. The Attributive data model represents all of the information accessed by application programs 130, 132 and 134 as various combinations of a single primitive. In general, application programs 130, 132 and 134 have their own databases which contain data either in the Attributive data model's format or in some other format. If the application program information is not in the Attributive data model format, it is converted into that format by means described below. In this manner, data structure 140 can be accessed by each of the application programs 130, 132, and 134.

Lowry also teaches (See col.9, lines 29-52 and col.10, lines 16-26) that Retrieval program 590, which behaves like the claimed "filter objects" retrieves data from the common databases 510 and 520, translates them into the target application 560 format, and inputs the revised target object data into the target application 560. Lowry does not expressly use the terms "target geometry" and "source geometry", however, it is inherent that both CAD and CAM files are comprised of geometrical information.

Lowry also teaches that the attribute data objects are mapped hierarchically (Lowry: col.2, line 60 to col.3, line 10. Emphasis Added):

Specifically, the method of creating the data structure comprises the steps, executed by said data processor, of creating a different one of the attribute data objects for each of the node data and organizing the attribute data objects hierarchically according to a being-held relationship by choosing from the attribute data objects for each of the attribute data objects, a holder data object such that a hierarchical being-held relationship exists between each attribute data object and a single holder data object.

28. In regards to Claim 8, Lowry expressly teaches the following limitations:

8. The method of Claim 7, wherein the step of generating the set of tree objects includes the steps of:

translating the one or more source properties to one or more target properties;

generating one or more modifier stacks, wherein the one or more modifier stacks are based on the one or more target properties; and

inserting the one or more modifier stacks into the hierarchical tree structure.

Lowry teaches the following (Lowry: col.5, lines 32-45):

In accordance with the present invention, application programs 130, 132, and 134 share a common data structure 140 which is based upon an Attributive data model. The Attributive data model represents all of the information accessed by application programs 130, 132 and 134 as various combinations of a single primitive. In general, application programs 130, 132 and 134 have their own databases which contain data either in the Attributive data model's format or in some other format. If the application program information is not in the Attributive data model format, it is converted into that format by means described below. In this manner, data structure 140 can be accessed by each of the application programs 130, 132, and 134.

Lowry also teaches (See col.9, lines 29-52 and col.10, lines 16-26) that Retrieval program 590, which behaves like the claimed "filter objects" retrieves data from the common databases 510 and 520, translates them into the target application 560 format, and inputs the revised target object data into the target application 560. Lowry does not expressly use the terms "target geometry" and "source geometry", however, it is inherent that both CAD and CAM files are comprised of geometrical information.

Lowry also teaches that the attribute data objects are mapped hierarchically (Lowry: col.2, line 60 to col.3, line 10. Emphasis Added):

Specifically, the method of creating the data structure comprises the steps, executed by said data processor, of creating a different one of the attribute data objects for each of the node data and organizing the attribute data objects hierarchically according to a

being-held relationship by choosing from the attribute data objects for each of the attribute data objects, a holder data object such that a hierarchical being-held relationship exists between each attribute data object and a single holder data object.

29. In regards to Claim 9, it differs from Claim 1 in that “source object” and “target object” are referred to as “first object” and “second object”, respectively.

Moreover, the final step is not referred to as “revising the target object”, but rather as “performing a third modification to the second object”. The rejection applied to Claim 1 applies to Claim 9 as well.

30. In regards to Claim 10, Lowry expressly teaches the following limitations:

10. The method of Claim 9, wherein the step of performing the first modification to the second object includes the step of performing a type of modification that cannot be performed using said first application.

It is inherent that functions performed in CAM programs are not supported by CAD programs. (See Lowry, col.1, lines 20-56). CAD programs are for component designs, while CAM programs are for automating the machining functions needed to produce the CAD designs. For further evidence of this, see the figure in the reference by David Bak (“CAD/CAM comes to Windows”), as well as the first two pages of the reference by Tom Beard (“Tying It All Together”). Therefore a modification performed in a CAM application cannot be performed in a CAD application, so modifications in target application 560 cannot be performed in source application 530.

31. In regards to Claim 12, it differs from Claim 1 in that it claims a

“computer-readable medium carrying one or more sequences of instructions” for

Art Unit: 2123

implementing the method of Claim 1, wherein the execution is performed by one or more processors. Lowry anticipates this as well, as shown in Fig.1.

32. In regards to Claim 13, it differs from Claim 1 in that it claims a system for implementing the method of Claim 1, wherein the system comprises "a memory; one or more processors coupled to memory; and a set of computer instructions contained in the memory ..." Lowry anticipates this as well, as shown in Fig.1.

Claim Rejections - 35 USC § 103

33. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

34. The prior art cited is as follows:

35. U.S. Patent 4,864,497. (Henceforth referred to as "**Lowry**").

36. Barequet et al. "A Data Front-End for Layered Manufacturing", Proceedings of 13th Annual Symposium on Computational Geometry. 1997. pp.231-239.

(Henceforth referred to as "**Barequet**")

37. Claims 3, and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lowry in view of Barequet.

38. In regards to Claim 3, Lowry expressly teaches the following limitations:

3. The method of Claim 1, wherein:

Art Unit: 2123

the source application is a Computer Aided Design (CAD) application;

the step of generating the source object in the source application includes the step of generating a CAD object in said CAD application;

the step of performing a second modification to said source object includes the step of performing a modification to the CAD object; and

However, while Lowry teaches the translation and shared use of files by CAD and CAM applications, Lowry does not expressly teach the translation and shared use of files by CAD and "rendering applications", as claimed in the following limitations:

the target application is a rendering application; and wherein

the step of translating the source object to the target object includes the step of translating the CAD object into a rendering object;

the step of performing the first modification to the target object includes the step of performing a modification to the rendering object;

the step of revising said target object includes the step of revising the rendering object to reflect the second modification that was made to the CAD object without undoing the first modification to the rendering object.

Barequet, on the other hand, teaches the use of a rendering application to view and edit CAD files. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lowry with the teachings of Barequet in order to enable the shared use and translation of files between CAD and "rendering applications", because the rendering application taught in Barequet is used to prepare files for CAM applications, and would be functionally equivalent to a CAM application in Lowry's invention. Barequet teaches: "The DFE (Data Front-End) system described in this paper is a geometric software package which supports all stages of preparing CAD data files for production." (Barequet, p.232, col.2, para.3).

39. In regards to Claim 11, Lowry expressly teaches the following limitations:

11. The method of Claim 9, wherein:
 - the first application is a Computer Aided Design (CAD) application;
 - the step of generating the first object in the first application includes the step of generating a CAD object in said CAD application;
 - the step of performing a second modification to said first object includes the step of performing a modification to the CAD object; and

However, while Lowry teaches the translation and shared use of files by CAD and CAM applications, Lowry does not expressly teach the translation and shared use of files by CAD and "rendering applications", as claimed in the following limitations:

- the second application is a rendering application; and wherein
- the step of translating the first object to the second object includes the step of translating the CAD object into a rendering object;
- the step of performing the first modification to the second object includes the step of performing a modification to the rendering object;
- the step of performing the third modification to the second object includes the step of performing a third modification to the rendering object to reflect the second modification that was made to the CAD object without undoing the first modification to the rendering object.

Barequet, on the other hand, teaches the use of a rendering application to view and edit CAD files. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lowry with the teachings of Barequet in order to enable the shared use and translation of files between CAD and "rendering applications", because the rendering application taught in Barequet is used to prepare files for CAM applications, and would be functionally equivalent to a CAM application in Lowry's invention. Barequet teaches: "The DFE (Data Front-End) system described in this paper is a

geometric software package which supports all stages of preparing CAD data files for production." (Barequet, p.232, col.2, para.3).

Response to Arguments

40. Applicants argue at p.8, paragraph 2 of the Appeal Brief (paper #28) that:

Lowry does not disclose or suggest that one of the applications 530 and 560 is capable of making a modification that is not supported by the other application.

Examiner respectfully disagrees. Lowry expressly teaches in the "Background of the Invention" (cols. 1-2) that the application programs that Lowry is integrating are CAD and CAM applications. It is inherent that CAM applications perform modifications that are not supported by CAD applications.

41. Applicants argue at p.9, paragraph 2 of the Appeal Brief (paper #28,) that:

Thus, the source and target objects are, generally, corresponding resources in different applications, rather than a shared resource.

Examiner agrees, and has interpreted the data in source application 530's internal database as being the source object, and the data in the target application 560's internal database as being the target object.

42. Applicants argue at p.10, paragraph 2 of the Appeal Brief (paper #28), emphasis added, that:

Keeping in mind that the original source object dictates the constitution of the original translated object by way of translation, three possible 'definitions' of the source object are hypothetically discussed, none of which will cause Claim 1 to be anticipated by Lowry. Specifically, scenarios are discussed, in the context of a database system as in Lowry, in which the source object is equated with:

1) a field of a record;

- 2) an entire record that includes a field that is modifiable and supported by only the target application; and
- 3) an entire record except for a field that is modifiable and supported only by the target application.

43. The Applicants argue three scenarios, that by Applicants' own admission are hypothetical scenarios, yet do not address the scenario that is expressly taught by Lowry.

Lowry expressly teaches that there are 3 different objects:

1. The data stored in the database of application program 530. Examiner has interpreted this as being the "source object".
2. The data stored in the common database.
3. The data stored in the database of application program 560. Examiner has interpreted this as being the "target object."

The common data structure represents all of the information accessed by the various application programs (See col.5, lines 32-59). In other words, once the source object is translated and stored in the common database, the fields accessed by the source program are a subset of the object stored in the common database. Likewise, the target program accesses a different subset of the fields of the object stored in the common database. Therefore Applicants' scenario 1 is not relevant, because it consists of more than one field of a record.

Lowry also expressly teaches the use of locks on (see col.11, lines 57-64 and col.35, lines 34-41) to ensure that two applications are not simultaneously modifying the same portion of the database, called a "macropage". Each "macropage" is a 64k byte portion of a file, where a file consists of 128 512-byte

pages (See col.12, lines 60-66). Locks are necessary because there are some fields that may be accessed by both the source and the target applications, and the locks are required to conflicts. The locks are required in the overlap between the subsets. It is when the target application modifies certain fields in the overlap, and the source application modifies others, that the claimed limitations are met.

Lowry also teaches that the programs being integrated include CAD and CAM applications (See col.1 and col.2). It is inherent that CAM applications perform functions that CAD applications do not. Therefore it is inherent that there are fields modifiable and supported only by CAM (target) application.

Therefore, the object stored in the common database fits the definition of scenario 2, and the source object does not. Therefore Applicants' scenario 2 is also not relevant.

44. Applicants argue at p.13, paragraph 1 of the Appeal Brief (paper #28) that:

However, if the original source object coming from the source application does not include the field that is modifiable and supported only by the target application, then the translated target object that corresponds to such a source object will not have that field either.

Examiner respectfully disagrees. The common data structure represents all of the information accessed by the various application programs (See col.5, lines 32-59). In other words, once the source object is translated and stored in the common database, the fields accessed by the source program are a subset of the object stored in the common database. Likewise, the target program accesses a different subset of the fields of the object stored in the common database. Therefore Applicants' argument in scenario 3 is incorrect.

45. Applicants also argue at p.13 of paragraph 3 of the Appeal Brief (paper #28) that:

Furthermore, regarding the 'snapshot' file 525' of Lowry, col.11, lines 52-55 states that the snapshot file is 'designed to represent a plurality of master data file 515'. Hence, if the Lowry database were used in an attempt to capture both the modification made by the source application and the modification made by the target application, there would be no single version of the master file that includes both of the modifications.

Examiner respectfully disagrees. The full text of the cited paragraph in

Lowry, (col.11, lines 47-57), is as follows:

Common data structure 140' in Fig.5B contains a master data file 515 and a snapshot file 525. Like slave database 520 of common data structure 140, snapshot file 525 of common data structure 140' contains at least one complete copy of a version of master data file 515. However, unlike slave database 520, snapshot file 525 is designed to represent a plurality of versions of master data file 515 by storing portions of master data file 515 which have been updated. In contrast, slave database 520 is designed to represent complete copies of the current version of master database 510.

Examiner also wishes to draw the Applicants' attention to the following paragraph in Lowry (col.9, lines 23-27. Emphasis added):

Preferably, the common data structure 140 shown in Fig.1 includes a master database 510 and a slave database 520. Any updates to the common data structure 140 are made to master database 510, and a copy of it is later made to form the next version of slave database 520.

The teachings in this last paragraph show that the Applicants' concern that "there would be no single version of the master file that includes both of the modifications" are unfounded. Examiner notes that both application programs 530 and 560 in Fig.5A retrieve their data from the slave database 520.

46. Finally, Applicants argue at p.14, paragraph 2 of the Appeal Brief (paper #28) that:

Art Unit: 2123

The Office Action contended that Applicants' remarks regarding the first scenario (scenario 1) of the previously-presented hypothetical are based on an assumption that 'the lock on the field in the target object is released before the source application modifies that object.' As is explained hereafter, the first scenario makes no such assumption.

Examiner refers the Applicants to an earlier paragraph in this section ("Response to Argument"), which shows that the entire argument based on scenario 1 is not relevant.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ayal I. Sharon whose telephone number is (703) 306-0297. The examiner can normally be reached on Monday through Thursday, and the first Friday of a biweek, 8:30 am – 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin Teska can be reached on (703) 305-9704. Any response to this office action should be mailed to:

Director of Patents and Trademarks
Washington, DC 20231

Hand-delivered responses should be brought to the following office:

4th floor receptionist's office
Crystal Park 2
2121 Crystal Drive
Arlington, VA

The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

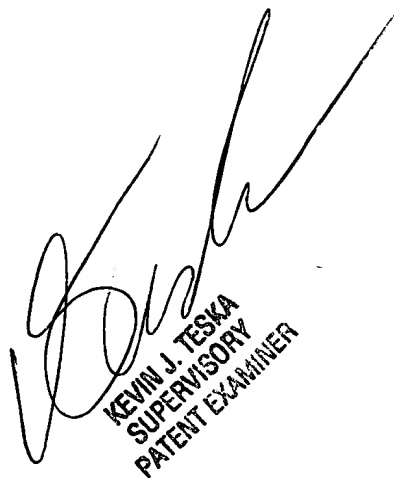
Art Unit: 2123

Any inquiry of a general nature or relating to the status of this application
or proceeding should be directed to the receptionist, whose telephone number is:
(703) 305-3900.

Ayal I. Sharon

Art Unit 2123

February 24, 2004



KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER